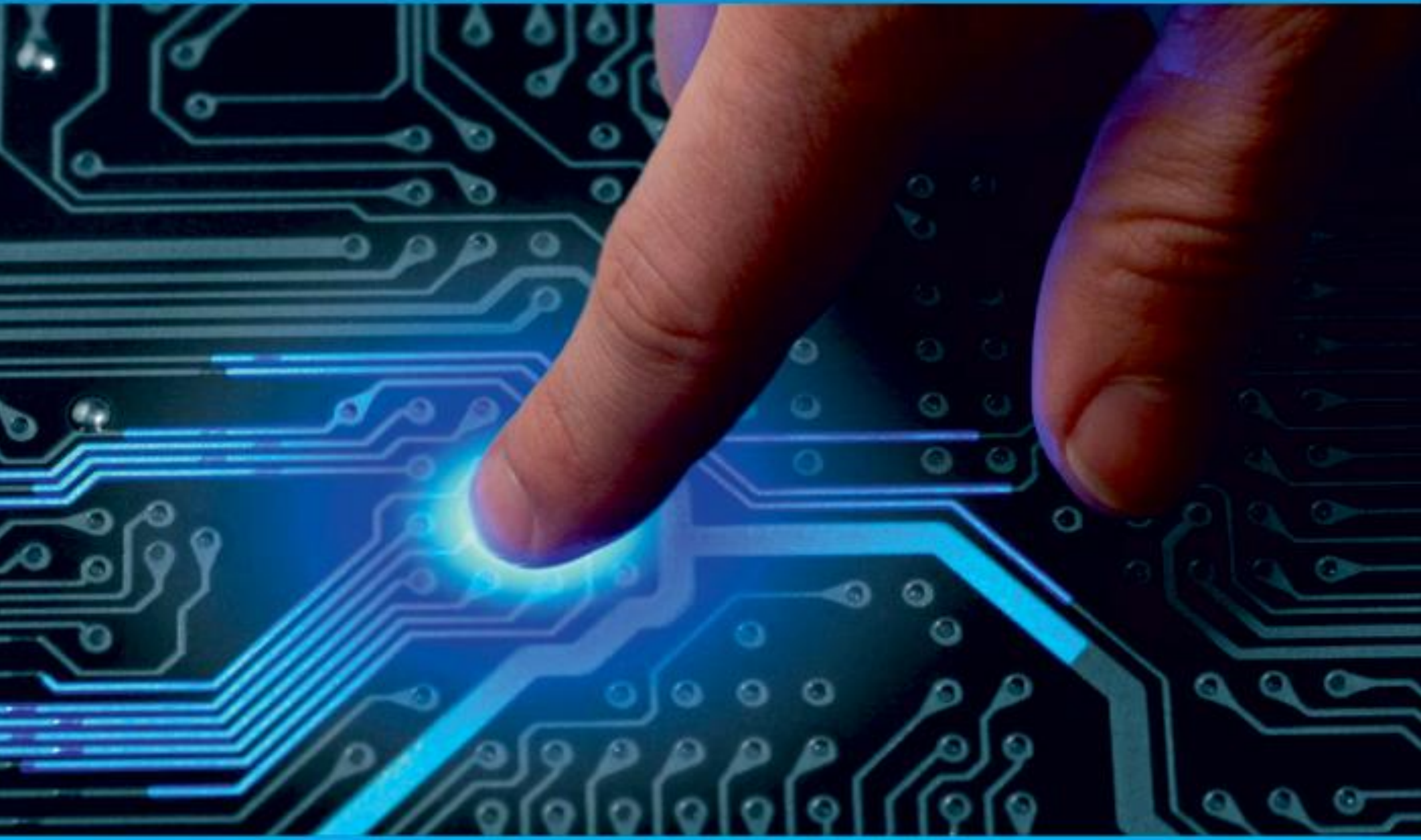
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204280 |

Collaborative Code Editor

Darshan G, Joel Sequiera, Darshan K, Jeevan Kumar J E, Shruthi K R

Department of Computer science and Engineering, Malnad College of Engineering, Hassan, Karnataka, India

ABSTRACT: In the realm of contemporary software engineering, the collaborative web code editor emerges as a pivotal tool in facilitating efficient teamwork and fostering creativity. This project endeavors to address the inherent challenges posed by traditional development environments by introducing a collaborative platform that transcends geographic boundaries and time constraints. Built upon the foundation of React.js and Node.js technologies, our collaborative web code editor enables real-time synchronization of code changes, empowering teams to work seamlessly on shared projects. Through features such as live code updates, simultaneous editing, and interactive communication channels, our platform facilitates dynamic collaboration and enhances productivity.

KEYWORDS: Real-time collaboration ,Code editing, Version control, Simplification, Automatic synchronization

I. INTRODUCTION

In the ever-evolving landscape of software engineering, collaboration has emerged as the linchpin of innovation and progress. As teams span continents and time zones, the imperative for seamless and efficient collaboration tools becomes increasingly pronounced. In response to this dynamic environment, we introduce a pioneering solution:
At the heart of our collaborative web code editor lies the transformative concept of real-time code synchronization. Unlike conventional development environments where code updates necessitate manual merging and synchronization, our platform ensures that every modification made by a user is instantly reflected across all connected devices.

Objectives

- Real-time Collaboration
- Enhanced Communication
- Scalability and Performance
- Intuitive User Interface

II. LITERATURE SURVEY

Aditya Kurniawan,Aditya Kurniawan,Christine Soesanto,Joe Erik Carla Wijaya; "CodeR: Real-time Code Editor Application for Collaborative"
The world of Internet is growing rapidly, many applications that previously created on the desktop start moving to the web. Many applications could be accessed anytime and anywhere easily using Internet. Developers need tools to create their applications, one of them named code editor. The purpose of this research is to design and develop a real-time code editor application using web socket technology to help users collaborate while working on the project. This application provides a feature where users can collaborate on a project in real-time. The authors using analysis methodology which conducting on a study of the current code editor applications, distributing questionnaires and conducting on literature study. CodeR is a web application that provides workspace to writing, perform, display the results of the code through the terminal, and collaborate with other users in real-time. The application main features are providing workspace to make, execute and build the source code, real-time collaboration, chat, and build the terminal.

Fiala, J; Yee-King, Matthew and Grierson, Mick. 2016. "Collaborative Coding Interfaces on the Web". *Proceedings of the International Conference on Live Interfaces*, The recent developments in Web technologies, including full-stack reactive application frameworks, peer-to-peer communication and client-side audiovisual APIs have introduced the possibility of creative collaboration in a number of contexts. Such technologies have the potential to transform the way Internet users interact with code. This paper introduces a theoretical and technical methodology for developing collaborative coding interfaces as web applications, tackling the issues of interactive rendering, user-platform interaction and collaboration. A number of existing interactive programming environments are reviewed, followed by a technical description and evaluation of C odeCircle, a collaborative coding web platform developed at Goldsmiths, University of London.

Rijul Saini Gunter Mussbacher; "Towards Conflict-Free Collaborative Modelling using VS Code Extensions" Model-Driven Engineering (MDE) advocates the use of models and their transformations, to better understand software systems and to increase the degree of automation across the software development process. However, with the increasing complexity of modern software systems, distributed development teams, and increasing time pressure for developing these systems, there is a need to collaborate more quickly when building and analyzing models. Furthermore, the COVID-19 pandemic has forced classroom-based software projects to organizational-level software systems to rely on virtual (web-based) collaborative development environments. Therefore, real-time collaborative modelling remains no longer an option but becomes a necessity for MDE too. In our previous work, we introduce a framework, tColab, which uses Eclipse Che workspaces to enable web-based collaborative modelling. However, with real-time collaboration, modelling conflicts can arise and their resolution goes beyond what is possible with the collaborative environment facilitated by an Eclipse Che workspace

Goldman, Max, Ph. D. Massachusetts Institute of Technology
"Software development with real-time collaborative editing" This thesis presents Collabode, a web-based integrated development environment for Java. With real-time collaborative editing, multiple programmers can use Collabode to edit the same source code at the same time. Collabode introduces error-mediated integration, where multiple editors see the text of one another's changes while being isolated from errors and in-progress work, and error-free changes are integrated automatically. Three models of collaborative programming are presented and evaluated using Collabode. Classroom programming brings zero-setup web-based programming to computer science students working in a classroom or lab. Test-driven pair programming combines two existing software development strategies to create a model with clear roles and explicit tool support. And micro-outsourcing enables one programmer to easily request and integrate very small contributions from many distributed assistants, demonstrating how a system for highly-collaborative programming enables a development model infeasible with current tools.

A. Design

The project aims to develop a collaborative web code editor with a robust frontend and backend architecture. Utilizing React.js for the frontend, the UI is designed to feature code editor windows, chat interfaces, and notifications, with React components organized for modularity and reusability. React's state management or Redux handles application state, including real-time updates. On the backend, Node.js with Express.js sets up a server for client requests and real-time communication, while MongoDB stores user data and collaboration history. WebSocket implementation via Socket.IO enables real-time collaboration, with features including code synchronization and conflict resolution. User authentication and authorization are implemented using Passport.js, with communication features like live chat and code commenting facilitating collaboration. Load balancing and caching ensure scalability and performance, while encryption and input validation maintain security and data integrity. Compatibility with major browsers and API integration are prioritized, alongside comprehensive documentation and testing to ensure code quality and functionality. The RCE platform provides users with a seamless environment for collaborative coding. Upon accessing the platform through a web browser or desktop application, users are presented with options to create or join coding rooms. As hosts, users can create rooms by generating unique room IDs and setting passwords, facilitating access for other participants. Alternatively, users can join existing rooms by entering the provided room ID and password. Once inside a room, users can engage in real-time collaborative coding using the integrated CodeMirror editor. This feature enables multiple users to write, edit, and debug code simultaneously, with changes instantly visible to all participants. Additionally, features like syntax highlighting, version control, and chat functionality further enhance the collaborative experience, fostering efficient communication and teamwork among coders.

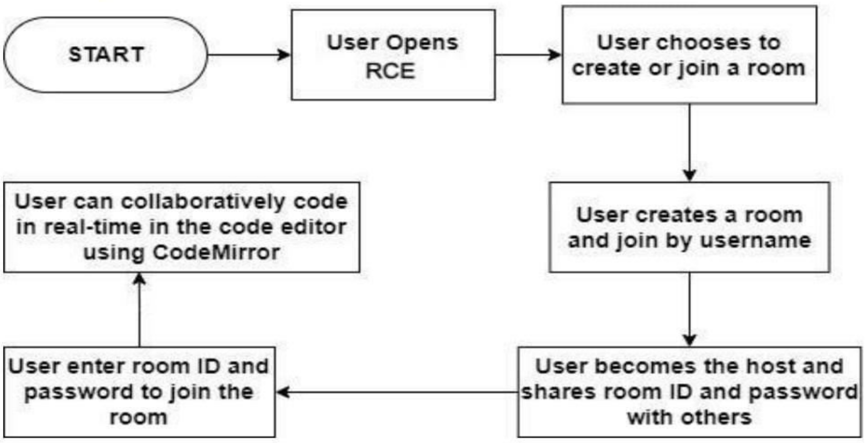

START
User Opens RCE
User chooses to create or join a room
User creates a room and join by username
User becomes the host and shares room ID and password with others
User enter room ID and password to join the room
User can collaboratively code in real-time in the code editor using CodeMirror

Figure1.High Level Design

B. Implementation

The implementation of the RCE platform begins with the development of a user-friendly interface accessible via web browsers or dedicated desktop applications. Users are presented with options to create or join coding rooms upon launching the platform. For room creation, hosts generate unique room IDs and set passwords, enabling access control for participants. Existing rooms are accessible through entering the provided room ID and password. Once inside a room, real-time collaborative coding is facilitated using the CodeMirror editor, allowing multiple users to write, edit, and debug code simultaneously. The integration of features like syntax highlighting, version control, and chat functionality enhances the collaborative experience, ensuring seamless communication and efficient teamwork. The backend architecture utilizes technologies like Node.js with Express.js for server setup and MongoDB for database management, enabling real-time communication and data storage. WebSocket implementation through libraries like Socket.IO enables bidirectional communication between clients and the server, supporting real-time collaboration feature .

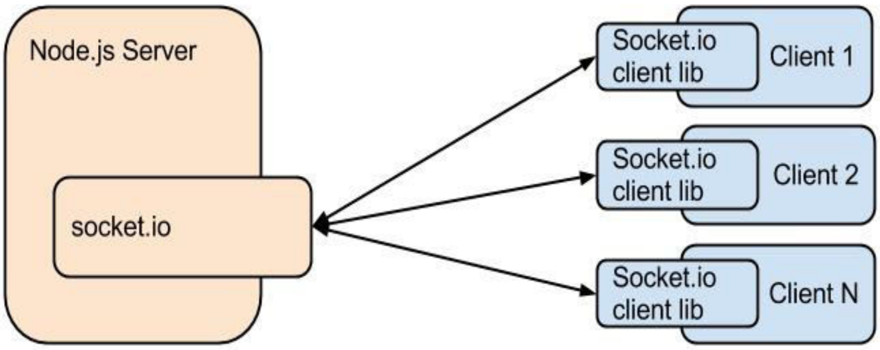


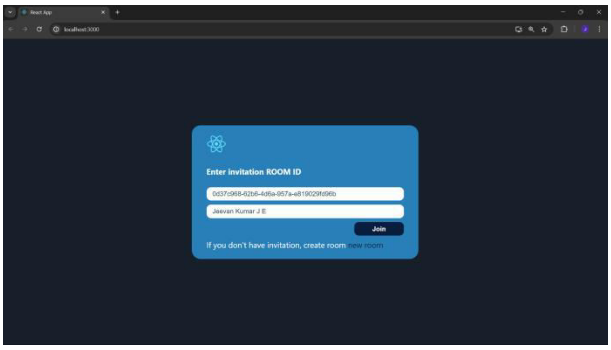
Figure2. socket.io Architecture

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204280 |

C. Results



Enter invitation ROOM ID

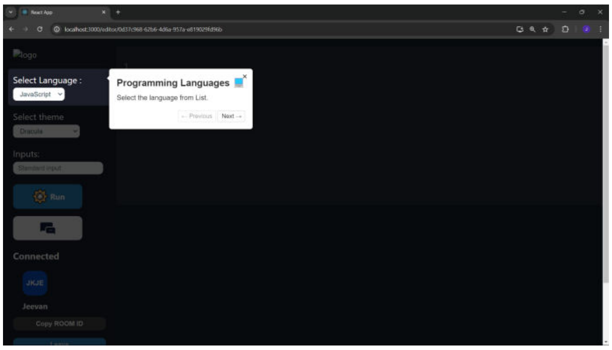
0d37c968-62b6-4d6a-957a-e819029fd96b

Jeevan Kumar J E

Join

If you don't have invitation, create room new room

Figure 3: Create/join room page



Select Language : JavaScript

Select theme: Dracula

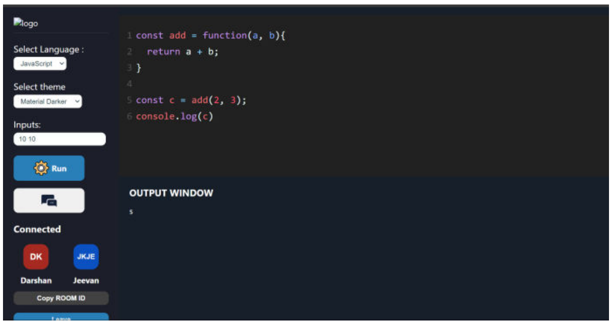
Inputs: Standard input

Programming Languages

Select the language from List.

← Previous | Next →

Figure 4: Page Quick Tour



Select Language : JavaScript

Select theme: Material Darker

Inputs: 10 10

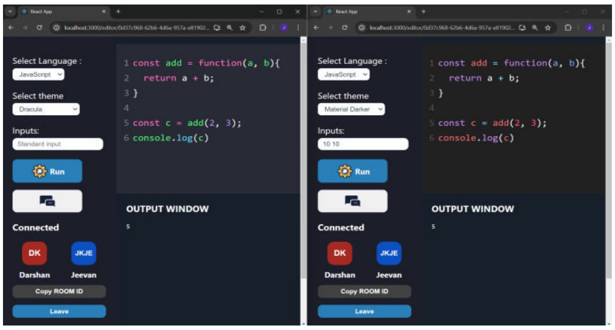
```
const add = function(a, b){
  return a + b;
}

const c = add(2, 3);
console.log(c)
```

OUTPUT WINDOW

```
5
```

Figure 5: Code Editor



Select Language : JavaScript

Select theme: Dracula

Inputs: Standard input

```
const add = function(a, b){
  return a + b;
}

const c = add(2, 3);
console.log(c)
```

OUTPUT WINDOW

```
5
```

Select Language : JavaScript

Select theme: Material Darker

Inputs: 10 10

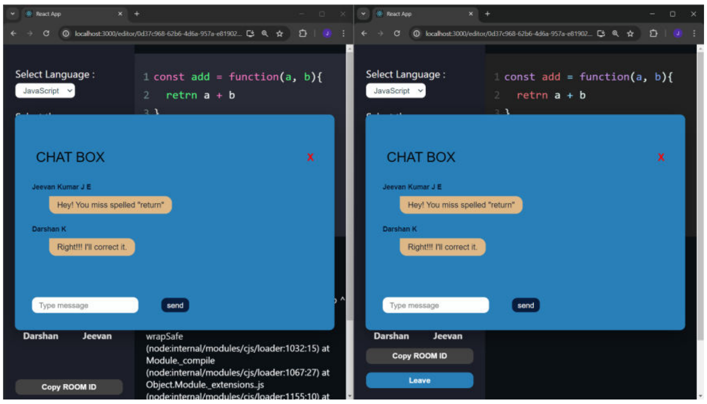
```
const add = function(a, b){
  return a + b;
}

const c = add(2, 3);
console.log(c)
```

OUTPUT WINDOW

```
5
```

Figure 6: Real Time Code Sync



Select Language : JavaScript

```
const add = function(a, b){
  retrn a + b
}
```

CHAT BOX

Jeevan Kumar J E: Hey! You miss spelled "return"

Darshan K: Right!!! I'll correct it.

Type message | send

```
wrapSafe
(node:internal/modules/cjs/loader:1032:15) at
Module._compile
(node:internal/modules/cjs/loader:1067:27) at
Object.Module._extensions..js
(node:internal/modules/cjs/loader:1155:10) at
```

Select Language : JavaScript

```
const add = function(a, b){
  retrn a + b
}
```

CHAT BOX

Jeevan Kumar J E: Hey! You miss spelled "return"

Darshan K: Right!!! I'll correct it.

Type message | send

Figure 7: Chat Window

III. CONCLUSION

In conclusion, the RCE platform offers a streamlined and intuitive solution for collaborative coding, catering to the needs of modern development teams. By allowing users to create or join coding rooms and seamlessly collaborate in real-time through features like the CodeMirror editor and integrated chat functionality, RCE facilitates efficient communication and code sharing. The platform's emphasis on security, with measures such as user authentication and secure room access, ensures the confidentiality and integrity of shared code. Overall, RCE simplifies the collaborative coding process, empowering teams to work together seamlessly and accelerate their development workflows.

REFERENCES

1. Aditya Kurniawan,Aditya Kurniawan,Christine Soesanto,Joe Erik Carla Wijaya; “CodeR: Real-time Code Editor Application for Collaborative”
2. Fiala, J; Yee-King, Matthew and Grierson, Mick. 2016. Collaborative Coding Interfaces on the Web. *Proceedings of the International Conference on Live Interfaces*,
3. Rijul Saini Gunter Mussbacher; “Towards Conflict-Free Collaborative Modelling using VS Code Extensions”
4. Goldman, Max, Ph. D. Massachusetts Institute of Technology “Software development with real-time collaborative editing”

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING